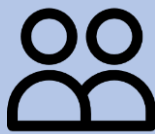


Taller en Sala Nro. 13 Recorridos sobre Grafos



En la vida real, los algoritmos de recorridos de grafos se utilizan para calcular las rutas en sistemas de mapas y en videojuegos, así http://kevanahlquist.com/osm_pathfinding/



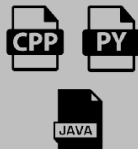
Trabajo en
Parejas



Hoy, plazo
máximo de
entrega



Docente entrega
código suelto en
GitHub



Sí .cpp, .py
o .java



No .zip, .txt,
html o .doc



Alumnos
entregan
código suelto
por GitHub

Ejercicios a resolver

1. Escriba el código para imprimir los vértices de un grafo en DFS. Suponga que existe una clase Graph con métodos getSuccessors(int i) y size(). El método g.GetSuccessors retornat los vecinos (o adyacentes) del vértice i
2. Escriba el código para imprimir los vértices de un grafo en BFS. Suponga que existe una clase Graph con métodos getSuccessors(int i) y size(). El método g.GetSuccessors retornat los vecinos (o adyacentes) del vértice i.
3. Usando DFS, escriba una implementación que retorne si hay camino o no entre un vértice i y un vértice j en un grafo g.
4. Usando BFS, escriba una implementación que retorne si hay camino o no entre un vértice i y un vértice j en un grafo g.

Ayudas para resolver los Ejercicios

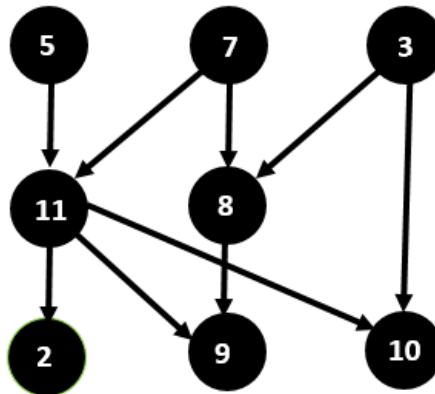
Ayudas para el Ejercicio 1.....	<u>Pág. 3</u>
Ayudas para el Ejercicio 2.....	<u>Pág. 5</u>
Ayudas para el Ejercicio 3.....	<u>Pág. 5</u>
Ayudas para el Ejercicio 4.....	<u>Pág. 5</u>

Ayudas para el Ejercicio 1



Como un ejemplo, para el grafo de la imagen, esta serían las respuestas:

Grafo



2	3	5	7	8	9	10	11
	8	11	8	9			2
	9	2	9				9
	10	9	11				10
		10	2				
			10				



Pista 1: Vea https://en.wikipedia.org/wiki/Depth-first_search#Pseudocode



Pista 2: Defina los métodos de esta forma:

```
public void dfs(Graph g, int v) {  
    boolean[] visitados = new boolean[g.size()];  
    dfsAux(Graph g, int v, visitados);  
}  
  
public void dfsAux(Graph g, int v, boolean[] vi){  
    ...  
}
```



Pista 3: Ingrese a <https://www.cs.usfca.edu/~galles/visualization/DFS.html>. Dada la representación lógica de varios grafos, infiera el orden en que se visitan los nodos usando búsqueda en profundidad (DFS). Revise si su respuesta es correcta.



Error Común 1: No utiliza el arreglo de visitados

```
public void dfsMalo(Graph g, int v) {  
    dfsAux(Graph g, int v, visitados);  
}
```



Error Común 2: El método *getSucessors* retorna los vecinos de un nodo, es decir, los nodos adyacentes. No es un recorrido en profundidad.

```
public void dfsAuxMalo(Graph g, int v, boolean[] vi){  
    vi[v] = true;  
    System.out.println(v);  
    ArrayList<Integer> vecinos = g.successors(v);  
    for (Integer vecino: vecinos) {  
        dfsAux(g, vecino, vi);  
    }  
}
```

Ayudas para el Ejercicio 2



Pista: Vean https://en.wikipedia.org/wiki/Breadth-first_search#Pseudocode

Ayudas para el Ejercicio 3



Pista: Vean <http://www.geeksforgeeks.org/find-if-there-is-a-path-between-two-vertices-in-a-given-graph/>



Error Común 1: No utiliza la respuesta que entrega el llamado recursivo

```
boolean dfsAuxMalo(Graph g, int v, boolean[] vi, int buscado) {  
    vi[v] = true;  
    if (v == buscado) return true;  
    ArrayList<Integer> vecinos = g.getSuccessors(v);  
    for (Integer vecino : vecinos)  
        if (!vi[vecino]) // !(vi[vecino] == true)  
            dfsAux(g, vecino, vi, buscado);  
    return false;  
}
```

Ayudas para el Ejercicio 4



Pista: Vean <http://www.geeksforgeeks.org/find-if-there-is-a-path-between-two-vertices-in-a-given-graph/>



Como un ejemplo para el grafo de la imagen del punto 1, estas serían las respuestas:

2 → 2
3 → 3, 8, 10, 9
5 → 5, 11, 2, 9, 10
7 → 7, 11, 8, 2, 9, 10
8 → 8, 9
9 → 9
10 → 10
11 → 11, 2, 9, 10



Pista: Ingrese a <https://www.cs.usfca.edu/~galles/visualization/BFS.html>. Dada la representación lógica de varios grafos, infiera el orden en que se visitan los nodos usando búsqueda en amplitud (BFS). Revise si su respuesta es correcta.



Error Común: Solo sabe usar BFS



¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 **Ext.** 9473

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agende una cita con él a través de <http://bit.ly/2gzVg10> , en la pestaña *Semana*. Si *no da clic en esta pestaña, parecerá que toda la agenda estará ocupada.*